What is claimed is:

1. A method for controlling a transaction flowing from a client process to a downstream process, the method comprising the steps of:

detecting a connection request, associated with the transaction, by a client server;

inputting said transaction from the client process into a client channel, said client server generating a client interface for said client channel upon detecting the connection request;

assigning a priority to said transaction by an input handler;

selectively loading the transaction from the client interface into a set of priority queues based on the priority assigned;

serializing, logging and routing said transaction flowing from said priority queues by a mapper;

communicating said transaction to an appropriate downstream process by the downstream interface; and,

coordinating connection requests and downstream process backflow messages by said downstream server.

2. The method as set forth in claim 1, further comprising the step of programming the input handler with metadata containing routing and identifying information to direct the transaction to a downstream process.

3. The method as set forth in claim 1, further comprising the step of detecting connection requests and transferring said transaction to and from said client processes.

4. The method as set forth in claim 1, further comprising the step of, at selected times, directly sending the transaction to the downstream process.

5. The method as set forth in claim 1, further comprising the step of adapting

an interface for performing programmed functions among the client server, the client interface, and the input handler, for establishing bi-directional connectivity and passing data.

6. The method as set forth in claim 1, further comprising the step of creating a transaction-record to store all metadata of said transaction, including a pointer to each transaction.

7. The method as set forth in claim 1, further comprising the step of assigning a priority to the transaction by the input handler, such priority being used to control latency, throughput of transactions, and placement of the transaction into the priority queue.

8. The method as set forth in claim 1, further comprising the step of replicating an additional mapper.

9. The method as set forth in claim 1, further comprising the step of running a repetitive algorithm in the mapper which selects transactions from the set of priority queues by revisiting each smaller numbered queue before proceeding to the next smaller numbered queue.

10. The method as set forth in claim 1, further comprising the step of replaying said transaction, from the log, upon a system fault and/or a request of a downstream process.

11. The method as set forth in claim 1, further comprising the step of the downstream server routing a backflow massage for communicating with the client process.

12. The method as set forth in claim 1, further comprising the step of broadcasting said transaction to multiple downstream processes.

13. The method as set forth in claim 1, further comprising the step of generating additional interfaces, while the system is in a real time run mode.

14. An apparatus for control of a transaction flowing from a client process to a downstream process, the apparatus comprising:

a client server configured to recognize a connection request from the client process;

a client channel for input of said transaction from the client process;

a client interface generated by said client server for accepting said transaction from the client channel upon detection of a connection request;

an input handler operative to direct the flow of the transaction;

a plurality of priority queues operative to store the transaction;

a mapper operative to serialize, log and route the transaction from the priority queues;

a downstream server which generates a downstream interface, said transaction being routed by the mapper to the downstream interface; and,

said downstream server configured to coordinate connection requests and downstream process backflow messages.

15. The apparatus according to claim 14, wherein each transaction contains metadata information, said metadata having routing and identification information to direct the transaction to a downstream process.

16. The apparatus according to claim 14, wherein a communication interface is used by the client server and the client interface, to detect connection requests and transfer said transaction to said client processes.

17. The apparatus according to claim 14, wherein at selected times, the client interface directly sends the transaction to the downstream process.

18. The apparatus according to claim 14, wherein an adaptive interface performs programmed function operations between the client server, the client interface, the communication interface and the input handler for establishing bi-directional connectivity and passing of said transaction.

19. The apparatus according to claim 14, wherein a transaction-record is created to store all metadata of said transaction and assigns a pointer to each transaction.

20. The apparatus according to claim 14, wherein a priority is assigned to each transaction by an input handler and is used to control latency, throughput of said transaction and placement of the transaction into the priority queue.

21. The apparatus according to claim 14, wherein at least one additional mapper is replicated when said transaction exceeds a bandwidth of the mapper.

22. The apparatus according to claim 14, wherein a repetitive algorithm processes the set of priority queues by revisiting each smaller numbered queue before proceeding to the next set.

23. The apparatus according to claim 14, wherein the downstream interface replays said transaction, from the log, upon a system fault and/or detection of corrupted data.

24. The apparatus according to claim 14, wherein the downstream server routes the backflow message for communicating with the client process.

25. The apparatus according to claim 14, wherein said transaction is broadcast to multiple downstream processes.

26. The apparatus according to claim 14, wherein additional interfaces are generated while the system is in a real time run mode.

27. An adaptive interface apparatus which performs programmed function operations for routing transactions in a data collection system, the apparatus comprising:

a client server operative to search for and establish connections to a client process;

a client interface generated by said client server upon detecting a connection request operative to establish a connection for the transfer of transactions;

a communication interface operative to detect a connection request, communicate with the client server, route said transactions to and from the client process and communicate with the client interface to fetch and send data over a client channel; and

an input handler operative to direct the flow of the transactions and communicate with the communication interface and the client interface.

28. The apparatus according to claim 27, wherein said client interface continuously cycles, searches and reads transactions from a client channel.

29. The apparatus according to claim 27, wherein pointers are assigned to the transactions and are passed to the functions needed for retrieving transactions from a client channel.

30. The apparatus according to claim 29, wherein said pointers are used as a reference between the input handler, the communications interface and the associated client interface.

31. The apparatus according to claim 27, wherein the communications interface operates at a low level and performs the functions of establishing links, retrieving data and sending data.

32. The apparatus according to claim 30, wherein the input handler prioritizes the transactions and determines the sequence for routing the transactions.

33. The apparatus according to claim 30, wherein a new client interface is generated, in real time run mode, upon detecting a new connection request.